Relatório de atividades do dia 27/03/2025

Aluno: João Pedro de Oliveira dos Santos Costa

RGM: 8830589887

1. O que são containers e porque usamos Docker para desenvolvimento?

*Os containers são uma tecnologia de virtualização, onde, suas unidades executáveis de software, empacotam o código, suas bibliotecas e dependências. Possibilitam que o código seja executado em qualquer ambiente de computação, seja desktop, TI tradicional ou Infraestrutura em nuvem.*

*Eles utilizam uma forma de virtualização do Sistema Operacional (SO), na qual recursos do kernel podem ser usados para isolar processos e controlar a quantidade de CPU, memória e disco que esses processos podem acessar.*

*Resumidamente, o Docker facilita a criação, implantação e gestão de aplicações.*

1. Qual a função do PHP-FPM e porque ele é utilizado Junto ao Nginx?

*O PHP-FPM (FastCGI Process Manager) é um serviço que é consideravelmente mais rápido que os outros métodos de se processar scripts em PHP, e também é escalável, permitindo o recebimento de mais requisições.*

*Ele é utilizado junto ao Nginx porque diferente do Apache, não possui módulo nativo para interpretar o PHP, assim, o Nginx encaminha as requisições para o PHP-FPM que processa e retorna para o Nginx.*

1. Qual a diferença entre um servidor Apache e um servidor Nginx?

*O Apache e o Nginx são dois servidores webs que possuem arquiteturas diferentes que os fazem ter especificidades.*

*O Apache possui uma arquitetura baseada em threads/processos onde cada requisição gera um processo novo, ele suporta o .htaccess, permitindo configurações por diretório e junto a isso, tem grande facilidade em rodar aplicações PHP pois pode rodar o PHP nativamente com o mod\_php.*

*Já o Nginx é baseado em eventos assíncronos, um único processo gera múltiplas conexões, sendo assim, ele lida bem com alto tráfego e concorrência, porém, ele precisa de um interpretador externo PHP-FPM. É recomendado para sites de tráfego alto e APIs escaláveis.*

1. Como o Docker organiza imagens e containers? Explique o conceito de multi-stage builds.

*As imagens no Docker são read-only e armazenadas em camadas sobrepostas (UnionsFS). Cada imagem é construída a partir de um Dockerfile e contém tudo o que um aplicativo precisa para rodar, como:*

*Sistema Operacional mínimo, bibliotecas e dependências, código da aplicação, configuração inicial, etc…*

*O Docker utiliza um cache local (/var/lib/docker) para armazenar imagens baixadas ou construídas, as imagens podem ser obtidas de registros como Docker Hub, Amazon ECR, Google Container Registry, etc.*

*Já a organização dos containers funciona da seguinte forma, um container é uma instância em execução de uma imagem Docker. Diferente da imagem, ele pode armazenar mudanças no estado da aplicação.*

*Cada container possui Id único, um nome, um filesystem e um ambiente de execução isolado. Os containers são isolados, mas podem compartilhar redes e volumes.*

*O multi-stage build é uma técnica docker que permite criar imagens mais leves e otimizadas ao dividir o processo de build em múltiplas etapas dentro de um único Dockerfile. A ideia é dividir o Dockerfile em múltiplas fases.*